



Java: vantaggi e applicazioni di un linguaggio moderno in contesti aziendali e produttivi

Presentazione a cura di:

Andrea Bertolini

Obiettivi seminario

- Overview Java, origini
- Vantaggi e svantaggi
- Applicazioni reali

Origini (curiosità)

- Nato nel ...
 - [A] 1990
 - **[B] 1995**
 - [C] 2000
 - [D] 2005
 - [E] 2015
- Obiettivo primario del Java Project
 - [A] Per comunicazioni universitarie
 - [B] Per videogiochi
 - **[C] Per televisione interattiva**
 - [D] Per applicazioni governative
 - [E] Per applicazioni su browser
- Origine del nome Java? Java Coffee
- Chi ha originariamente realizzato e distribuito Java?
Sun Microsystem

Motivi del successo

- Write Once, Run Anywhere (WORA)
- Applet – web interattivo
- Licenza GPL per JVM (2006/2007)
- Java Community Process (JCP) (1998)

- Java SE (Standard Edition)
 - Per ambienti Server e Desktop
 - API di base per linguaggio Java
 - Implementazione più conosciuta: JDK
- Java EE (Enterprise Edition)
 - Per applicazioni web
 - API per sviluppo di architetture multi-tier e scalabili, web services, ...
 - Implementati Web Application Servers su questo standard
- Java ME (Micro Edition)
 - Per sistemi Embedded
 - IoT

- Rapida ed esponenziale evoluzione di Internet e del world-wide web
- Sempre maggiore richiesta di
 - Sicurezza (es. e-commerce)
 - Performance
 - Robustezza nelle applicazioni
 - Flessibilità (multi-piattaforma)

1. Simple, Object Oriented, and Familiar

- Semplice
 - da imparare
 - da utilizzare
 - da capire
- Orientato agli oggetti
 - dalla base
 - per applicazioni complesse
 - per struttura logica ordinata, pulita ed efficiente
- Familiare
 - mantenendo stessa struttura di C++
 - rimuovendo complessità non fondamentale
 - per rapida migrazione a questo linguaggio

2. Robust and Secure

- Robusto
 - controlli a più livelli (compilazione e runtime)
 - gestione della memoria semplice e interna a JVM (non richiesta gestione manuale)
 - per aumentare il numero di errori rilevati PRIMA
- Sicuro
 - Ideato per ambienti distribuiti
 - security features per controlli di sicurezza in runtime

3. Architecture Neutral and Portable

- Indipendente dall'architettura
 - ideato per
 - Infrastruttura rete eterogenea
 - Architetture hardware eterogenee
 - Essere eseguito su differenti sistemi operativi
 - Comunicare con interfacce di differenti linguaggi
 - Genera bytecodes per astrarsi dall'implementazione hardware
- Portabile
 - Definisce convenzioni per dimensione dei dati primitivi e operatori aritmetici
 - Programmi rimangono gli stessi ovunque
 - Evita incompatibilità di dati tra architetture hardware o software differenti
 - JVM: specifiche di macchina astratta per cui i compilatori generano i bytecodes

4. High Performance

- Performance
 - Garbage Collector come thread continuo a bassa priorità
 - Alta probabilità che memoria sia disponibile quando serve
 - Componenti computazionalmente complessi possono essere scritti in linguaggio nativo e interfacciati con Java
 - Alte performance nonostante sia linguaggio interpretato

5. Interpreted, Threaded, and Dynamic

- Interpretato
 - Interprete può eseguire direttamente bytecodes sulla macchina
 - Fase di linking è semplice, incrementale e leggera
 - Permette rapidi cicli di sviluppo / prototipazione
- Threaded
 - Necessità di eseguire più operazioni contemporaneamente
 - Supporto a multi-threading per attività concorrenti
 - Risulta in alti livello di interazione per l'utente
 - API per supporto a multi-threading con logiche thread-safe
- Dinamico
 - Classi linkate solo quando necessario
 - Moduli possono essere aggiunti in on-demand: servizi in costante evoluzione

Vantaggi

- Combinazione di tutti i principi elencati
- Basi solide: piattaforma evoluta e in costante miglioramento
- Produzione di software:
 - Portabile
 - con interfacce grafiche semplici da realizzare e utilizzare
 - Sicuro
 - con alte performance
- Possibilità di sviluppare software concentrandosi sull'obiettivo

Svantaggi (critiche più comuni)

- Mancanza di distruttori espliciti
- Garbage collector costoso e sviluppato male in alcune implementazioni
- Forte impatto sulla memoria per JVM
- Molto diffuso da molti anni: librerie in circolazione non sempre buone
- Molto di alto livello



Applicazioni in contesti reali (Infolog)



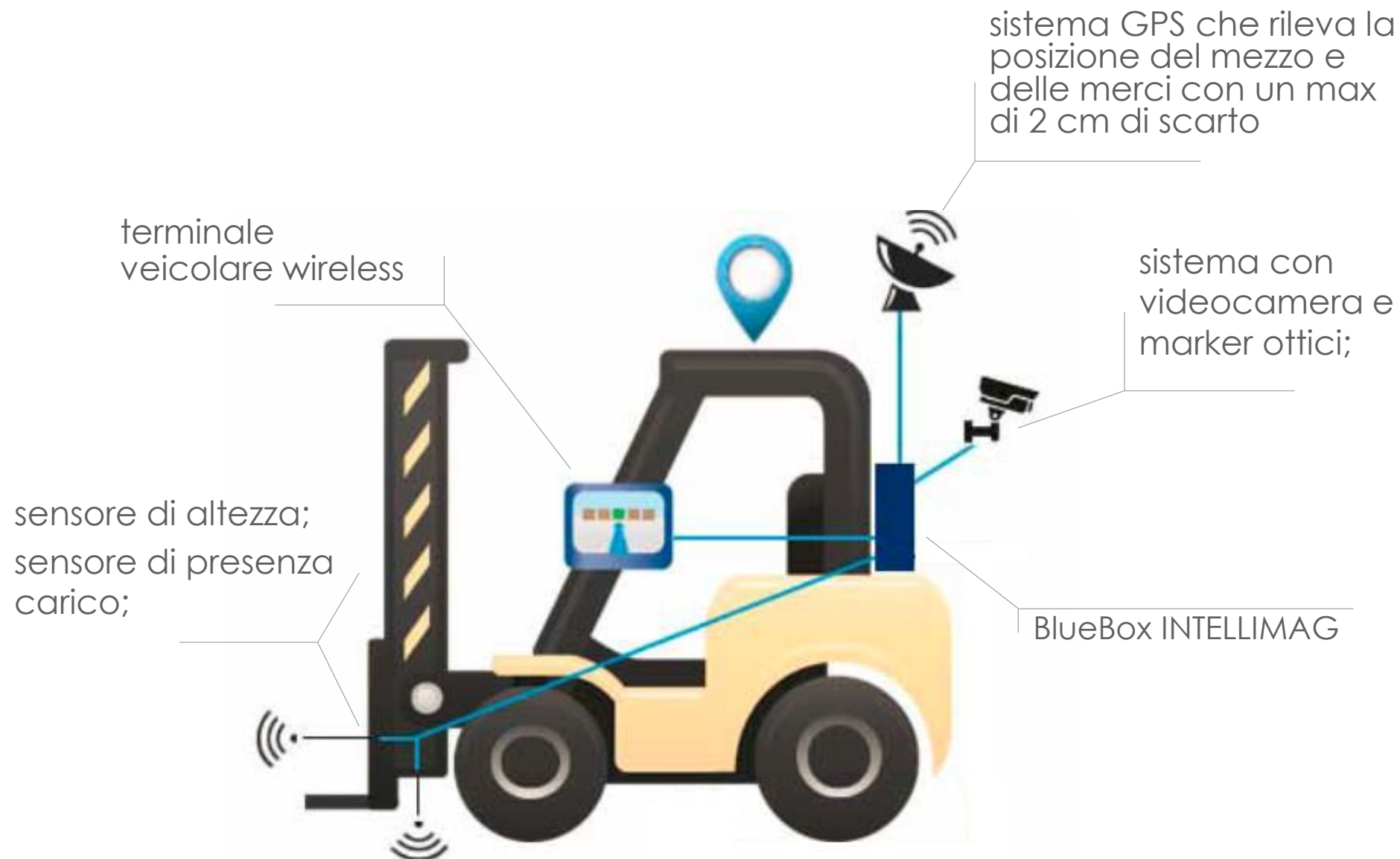
- Software proprietario per la **logistica**
- Ottimizza i flussi di magazzino evitando sprechi di risorse e una continua tracciabilità della merce
- Tramite il modulo di *regia attiva* permette massima efficienza
- Con l'utilizzo di moduli *tecnologici* e *innovativi* rende la logistica un settore all'avanguardia

Ricerca & Sviluppo

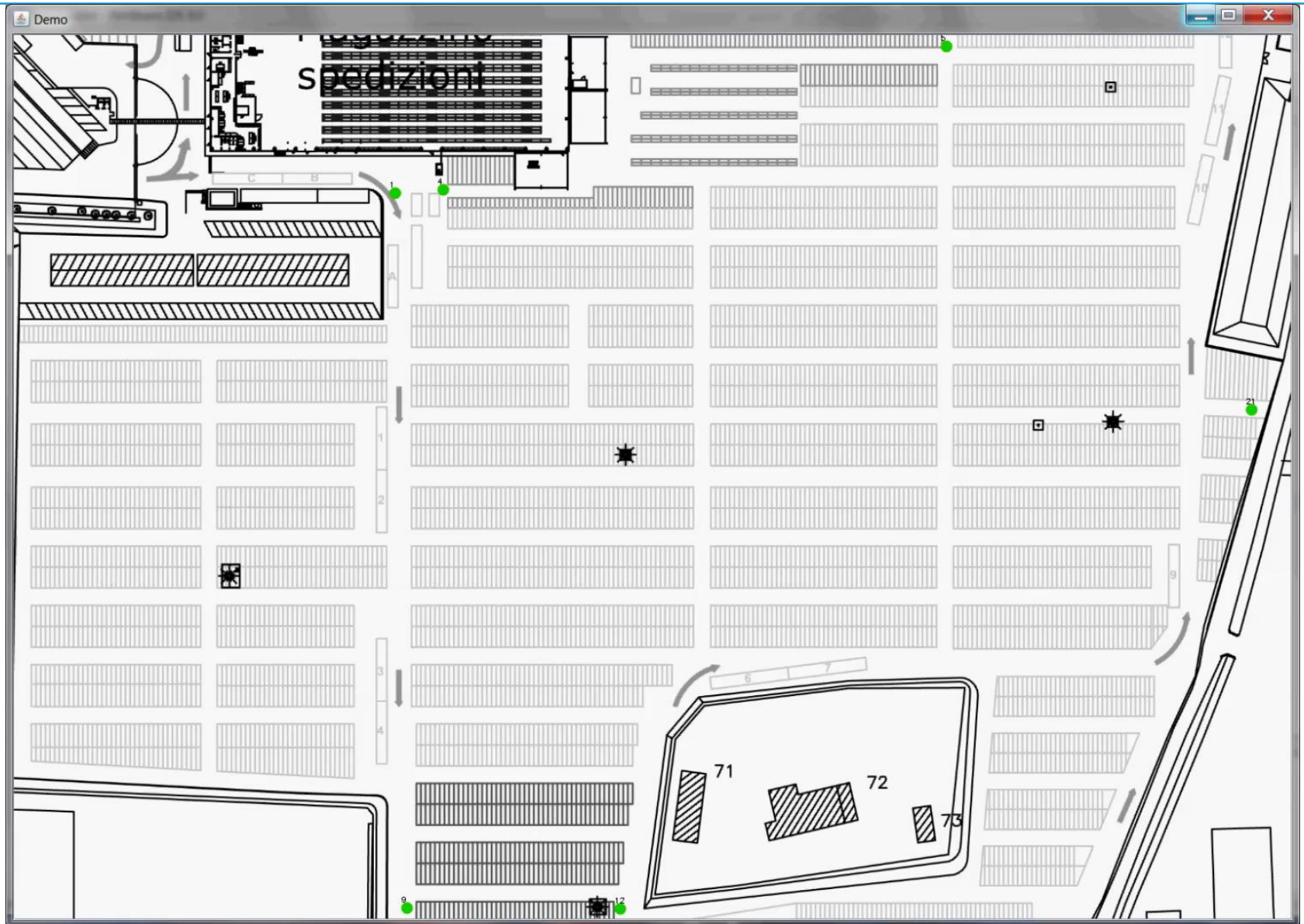
Interfacce

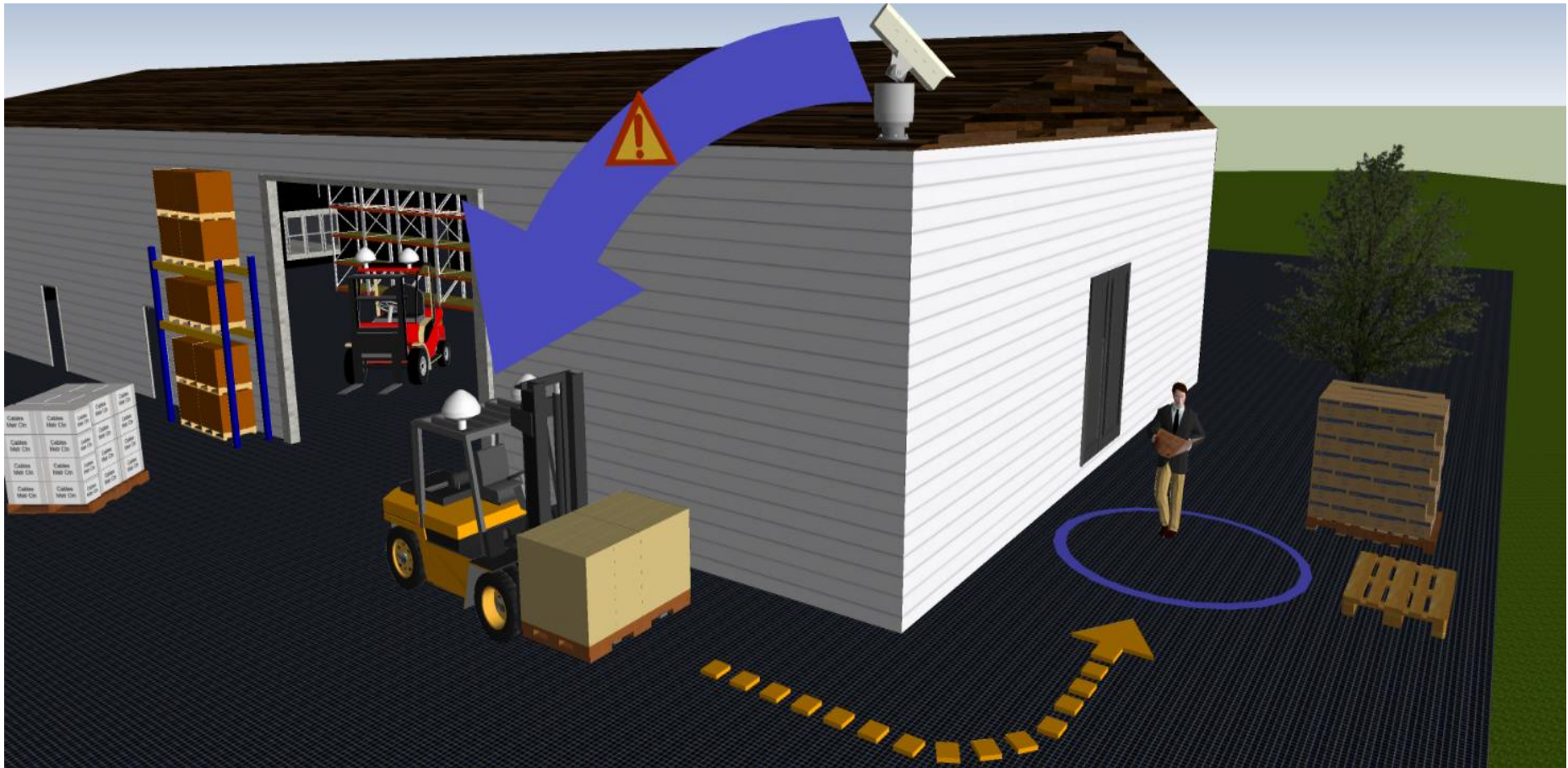


Tecnologia



Prevenzione collisioni









- Localizzazione di merce in magazzino coperto
- Tracciamento continuo di elementi dalla dimensione limitata
- Analisi del percorso per valutare il passaggio della merce da gate predefiniti

Java e Mobile

- Realizzate applicazioni Android e iOS per trasposizione di logiche di Intellimag
- Pannello di controllo per la situazione di magazzino
- Architettura snella e flessibile per operatività veloce



Grazie dell'attenzione

Modena – 09 dicembre 2015

INFOLOG GROUP
business solutions

INFOLOG SpA

via Pier Paolo Pasolini, 23

41123 Modena (Mo)

info@infolog.it - www.infolog.it

